eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Material de estudio

OCW 2019: *Curso práctico para el análisis e inferencia estadística con Mathematica*

Tema 4. Probabilidad y combinatoria

Equipo docente del curso
Arrospide Zabala, Eneko
Martín Yagüe, Luis
Unzueta Inchaurbe, Aitziber
Soto Merino, Juan Carlos
Durana Apaolaza, Gaizka
Bikandi Irazabal, Iñaki

Departamento de Matemática Aplicada
Escuela de Ingeniería de Bilbao, Edificio II-I

OCW OpenCourseWare

TEMA 4. PROBABILIDAD Y COMBINATORIA

Introducción

Función de probabilidad

Un experimento aleatorio es aquél del que se conocen sus posibles resultados y que, repetido en las mismas condiciones, no siempre produce los mismos efectos.

Un suceso elemental es cada uno de los resultados que se pueden obtener al realizar un experimento aleatorio. El conjunto de todos ellos se denomina espacio muestral y se denota Ω.

Una función de probabilidad, P, es la aplicación que hace corresponder a todos los sucesos de un espacio muestral uno y sólo un valor real perteneciente al intervalo $[0,1]$:

$$\boxed{P : \Omega \longrightarrow [0, 1] \subset \mathbb{R}}$$

Regla de Laplace

Cuando todos los sucesos de un espacio muestral son equiprobables puede calcularse la probabilidad de uno de ellos, A, dividiendo el número de casos favorables a la ocurrencia de A entre el número de casos posibles:

$$\boxed{P(A) = \frac{\text{número de casos favorables}}{\text{número de casos posibles}}}$$

Combinatoria

Por Análisis Combinatorio, o Combinatoria, se entiende aquella parte de las Matemáticas que se ocupa del estudio y propiedades de los grupos que pueden formarse con unos elementos dados, distinguiéndose entre sí por el número de elementos que entran en cada grupo y/o por el orden en que están dispuestos.

Según los criterios empleados para la formación, las agrupaciones pueden ser de tres tipos: variaciones, permutaciones y combinaciones.

Recuento de casos

El análisis combinatorio permite, en relación a un experimento aleatorio y sin perder ningún caso, el recuento de casos para aplicar la regla de Laplace.

Supóngase que un experimento aleatorio consta de r partes.
El principio básico de recuento generalizado determina que si en la primera parte se pueden obtener n_1 posibles resultados, n_2 en la segunda parte y, así sucesivamente, entonces existen un total de $n_1 \cdot n_2 \cdot \ldots \cdot n_r$ resultados posibles del experimento.

Permutaciones

Sin repetición

Permutaciones de n elementos son los diferentes grupos que pueden formarse con los n elementos de modo que dos grupos difieren entre sí porque sus elementos están dispuestos en diferente orden.

- **Permutations[*list*]**. Genera una lista que contiene todas las posibles permutaciones que pueden realizarse con los elementos de la lista que se facilita como argumento.

Cálculo: $\boxed{P_n = n!}$

- **Factorial[*n*]**. Calcula el factorial del número n. También, puede usarse **n!**.

Ejemplo. Obtención de todas las posibles permutaciones que pueden realizarse con las letras a, b y c. ¿Cuántas son?

- definición de la lista

```
listaabc = {a, b, c};
```

- posibles permutaciones

```
permabc = Permutations[listaabc]
```

```
{{a, b, c}, {a, c, b}, {b, a, c}, {b, c, a}, {c, a, b}, {c, b, a}}
```

- número de permutaciones
  - contando el número de elementos de la lista **permabc**

```
Length[permabc]
```

```
6
```

  - aplicando la fórmula

```
Factorial[Length[listaabc]]
```

```
6
```

- **DistinctPermutations[*list*]**. Función análoga a **Permutations**. Necesita la carga del paquete **Combinatorica**.

Ejemplo. Hallar todas las posibles permutaciones que pueden realizarse con las letras a, b, c y d. ¿Cuántas son?

- definición de la lista

```
listaabcd = {a, b, c, d};
```

- carga del paquete **Combinatorica**

```
In[11]:= Needs["Combinatorica`"]
```

- posibles permutaciones

```
permabcd = DistinctPermutations[listaabcd]
```

```
{{a, b, c, d}, {a, b, d, c}, {a, c, b, d}, {a, c, d, b}, {a, d, b, c}, {a, d, c, b}, {b, a, c, d}, {b, a, d, c},
 {b, c, a, d}, {b, c, d, a}, {b, d, a, c}, {b, d, c, a}, {c, a, b, d}, {c, a, d, b}, {c, b, a, d}, {c, b, d, a},
 {c, d, a, b}, {c, d, b, a}, {d, a, b, c}, {d, a, c, b}, {d, b, a, c}, {d, b, c, a}, {d, c, a, b}, {d, c, b, a}}
```

- número de permutaciones
  - contando el número de elementos de la lista **permabc**

```
Length[permabcd]
```

```
24
```

  - aplicando la fórmula

```
Factorial[Length[listaabcd]]
```

```
24
```

Con repetición

Sea un conjunto formado por n elementos de los cuales r son distintos. El primero aparece n_1 veces, el segundo n_2 veces y así, sucesivamente, hasta el r-ésimo que aparece n_r veces (donde $n = n_1 + n_2 + \ldots + n_r$). Permutaciones con repetición de esos n elementos son los grupos que pueden formarse de modo que dos grupos difieren entre sí porque sus elementos están dispuestos en diferente orden.

- **Permutations[*list*]**. Genera una lista que contiene todas las posibles permutaciones que pueden realizarse con los elementos de la lista que se facilita como argumento. Los elementos repetidos se consideran como idénticos.

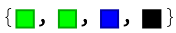
Cálculo: $\boxed{PR_n^{n_1,n_2,\ldots,n_r} = \dfrac{n\,!}{n_1\,!\cdot n_2\,!\cdot\ldots\cdot n_r\,!}}$ donde $n = n_1 + n_2 + \ldots + n_r$

- **Multinomial[*n1,n2, ...,nr*]**. Devuelve el coeficiente multinomial que permite el cálculo del número de permutaciones con repetición.

Ejemplo. Obtención de todas las posibles permutaciones que pueden realizarse con cuatro cuadrados de colores: dos verdes, uno azul y otro negro. ¿Cuántas son?

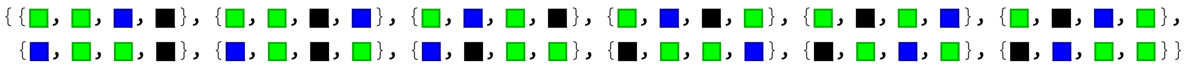
- definición de la lista

```
color = {Green, Green, Blue, Black}
```



- posibles permutaciones

```
permcolor = Permutations[color]
```



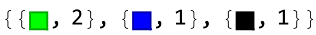
- número de permutaciones
  - contando el número de elementos de la lista **permcolor**

```
Length[permcolor]
```

12

  - aplicando la fórmula

```
n = Tally[color]
```



```
repcolor = Transpose[n][[2]] (* número de veces que se repite cada elemento *)
```

{2, 1, 1}

```
Multinomial[Apply[Sequence, repcolor]]
```

12

```
Factorial[Length[color]] / (Product[Factorial[repcolor[[i]]], {i, Length[n]}])
```

12

Ejemplo. Obtención de todas las posibles permutaciones que pueden realizarse con las letras de la palabra BARRA. ¿Cuántas son?

- definición de la lista

```
barra = {"b", "a", "r", "r", "a"};
```

- posibles permutaciones

```
permbarra = Permutations[barra]
```

```
{{b, a, r, r, a}, {b, a, r, a, r}, {b, a, a, r, r}, {b, r, a, r, a}, {b, r, a, a, r}, {b, r, r, a, a},
 {a, b, r, r, a}, {a, b, r, a, r}, {a, b, a, r, r}, {a, r, b, r, a}, {a, r, b, a, r}, {a, r, r, b, a},
 {a, r, r, a, b}, {a, r, a, b, r}, {a, r, a, r, b}, {a, a, b, r, r}, {a, a, r, b, r}, {a, a, r, r, b},
 {r, b, a, r, a}, {r, b, a, a, r}, {r, b, r, a, a}, {r, a, b, r, a}, {r, a, b, a, r}, {r, a, r, b, a},
 {r, a, r, a, b}, {r, a, a, b, r}, {r, a, a, r, b}, {r, r, b, a, a}, {r, r, a, b, a}, {r, r, a, a, b}}
```

- número de permutaciones
  - contando el número de elementos de la lista **permcolor**

```
Length[permbarra]
```

```
30
```

  - aplicando la fórmula

```
n = Tally[barra]
```

```
{{b, 1}, {a, 2}, {r, 2}}
```

```
repbarra = Transpose[n][[2]] (* número de veces que se repite cada elemento *)
```

```
{1, 2, 2}
```

```
Multinomial[Apply[Sequence, repbarra]]
```

```
30
```

```
Factorial[Length[barra]] / (Product[Factorial[repbarra[[i]]], {i, Length[n]}])
```

```
30
```

Variaciones

Sin repetición

Variaciones de n elementos tomados de r en r son los diferentes grupos que pueden formarse con los n elementos dados, tomados de r en r, de modo que dos grupos difieren entre sí bien porque tienen distintos elementos bien porque sus elementos están dispuestos en diferente orden.

Es un caso particular de permutaciones cuando no se consideran todos los elementos.

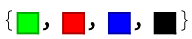
  - **Permutations[*list*,{*r*}]**. Genera una lista que contiene todas las posibles permutaciones que pueden realizarse con los elementos de la lista dada como primer argumento tomados de r en r.

Cálculo: $\boxed{V_{n,r} = n \cdot (n-1) \cdot (n-2) \cdot \ \ldots \ \cdot [n-(r-1)] = \frac{n!}{(n-r)!}}$

Ejemplo. Obtención de todas las posibles variaciones que pueden realizarse con cuatro cuadrados de diferentes colores tomados de dos en dos. ¿Cuántas son?

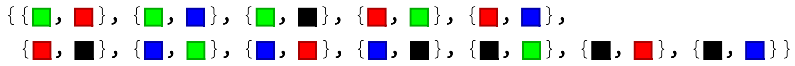
- definición de la lista

```
colorv = {Green, Red, Blue, Black}
```



- posibles variaciones

```
varcolor = Permutations[colorv, {2}]
```



- número de variaciones
    - contando el número de elementos de la lista **varcolor**

```
Length[varcolor]
```

```
12
```

- aplicando la fórmula

```
nv = Length[colorv]; rv = 2;
```

```
Factorial[nv] / Factorial[nv - rv]
```

```
12
```

Ejemplo. Obtención de todos los números de tres cifras que pueden escribirse con los dígitos del número 12.345. ¿Cuántas son?

- definición de la lista

```
numv = {1, 2, 3, 4, 5}
```

```
{1, 2, 3, 4, 5}
```

- posibles variaciones

```
varnum = Permutations[numv, {3}]
```

```
{{1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 2}, {1, 3, 4}, {1, 3, 5}, {1, 4, 2}, {1, 4, 3}, {1, 4, 5}, {1, 5, 2},
 {1, 5, 3}, {1, 5, 4}, {2, 1, 3}, {2, 1, 4}, {2, 1, 5}, {2, 3, 1}, {2, 3, 4}, {2, 3, 5}, {2, 4, 1}, {2, 4, 3},
 {2, 4, 5}, {2, 5, 1}, {2, 5, 3}, {2, 5, 4}, {3, 1, 2}, {3, 1, 4}, {3, 1, 5}, {3, 2, 1}, {3, 2, 4}, {3, 2, 5},
 {3, 4, 1}, {3, 4, 2}, {3, 4, 5}, {3, 5, 1}, {3, 5, 2}, {3, 5, 4}, {4, 1, 2}, {4, 1, 3}, {4, 1, 5}, {4, 2, 1},
 {4, 2, 3}, {4, 2, 5}, {4, 3, 1}, {4, 3, 2}, {4, 3, 5}, {4, 5, 1}, {4, 5, 2}, {4, 5, 3}, {5, 1, 2}, {5, 1, 3},
 {5, 1, 4}, {5, 2, 1}, {5, 2, 3}, {5, 2, 4}, {5, 3, 1}, {5, 3, 2}, {5, 3, 4}, {5, 4, 1}, {5, 4, 2}, {5, 4, 3}}
```

- número de variaciones
    - contando el número de elementos de la lista **varnum**

```
Length[varnum]
```

```
60
```

- aplicando la fórmula

```
nv = Length[numv]; rv = 3;
```

```
Factorial[nv] / Factorial[nv - rv]
```

```
60
```

Con repetición

Variaciones con repetición de n elementos tomados de r en r son los diferentes grupos que pueden formarse con los n elementos dados, tomados de r en r, en los que pueden aparecer elementos repetidos, de modo que dos grupos difieren entre sí bien porque tienen distintos elementos bien porque sus elementos están dispuestos en diferente orden.

- **Tuples[*list*,{*r*}]**. Genera una lista que contiene todas las posibles r-tuplas que pueden formarse con los elementos de la lista (que se pueden repetir) dada como primer argumento.

Cálculo: $\boxed{VR_{n,r} = n^r}$

- **Power[*x*,*y*]**. Calcula la potencia x^y. También, puede plantearse **x^y**.

Ejemplo. Obtención de todas las posibles variaciones con repetición que pueden realizarse con cuatro cuadrados de colores diferentes. ¿Cuántas son?

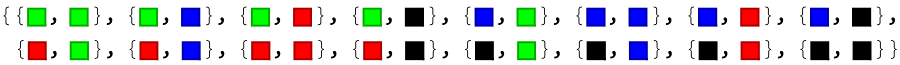
- definición de la lista

```
In[1]:= colorvr = {Green, Blue, Red, Black}
```

Out[1]= 

- posibles variaciones con repetición

```
In[2]:= varrcolor = Tuples[colorvr, {2}]
```

Out[2]= 

- número de variaciones
  - contando el número de elementos de la lista **varrcolor**

```
In[3]:= Length[varrcolor]

Out[3]= 16
```

  - aplicando la fórmula

```
In[4]:= nv = Length[colorvr]; rv = 2;

In[5]:= nv^rv

Out[5]= 16

In[6]:= Power[nv, rv]

Out[6]= 16
```

Ejemplo. Obtención de todos los números de tres cifras que pueden escribirse con los dígitos del número 11.335. ¿Cuántas son?

- definición de la lista

```
In[7]:= numvr = {1, 1, 3, 3, 5};
```

- posibles variaciones con repetición; debe tenerse en cuenta que, desde el punto de vista de la ordenación, los dígitos iguales se consideran elementos diferentes

```
In[8]:= Tuples[numvr, {3}]

Out[8]= {{1, 1, 1}, {1, 1, 1}, {1, 1, 3}, {1, 1, 3}, {1, 1, 5}, {1, 1, 1}, {1, 1, 1}, {1, 1, 3}, {1, 1, 3}, {1, 1, 5}, {1, 3, 1},
  {1, 3, 1}, {1, 3, 3}, {1, 3, 3}, {1, 3, 5}, {1, 3, 1}, {1, 3, 1}, {1, 3, 3}, {1, 3, 3}, {1, 3, 5}, {1, 5, 1}, {1, 5, 1},
  {1, 5, 3}, {1, 5, 3}, {1, 5, 5}, {1, 1, 1}, {1, 1, 1}, {1, 1, 3}, {1, 1, 3}, {1, 1, 5}, {1, 1, 1}, {1, 1, 1}, {1, 1, 3},
  {1, 1, 3}, {1, 1, 5}, {1, 3, 1}, {1, 3, 1}, {1, 3, 3}, {1, 3, 3}, {1, 3, 5}, {1, 3, 1}, {1, 3, 1}, {1, 3, 3}, {1, 3, 3},
  {1, 3, 5}, {1, 5, 1}, {1, 5, 1}, {1, 5, 3}, {1, 5, 3}, {1, 5, 5}, {3, 1, 1}, {3, 1, 1}, {3, 1, 3}, {3, 1, 3}, {3, 1, 5},
  {3, 1, 1}, {3, 1, 1}, {3, 1, 3}, {3, 1, 3}, {3, 1, 5}, {3, 3, 1}, {3, 3, 1}, {3, 3, 3}, {3, 3, 3}, {3, 3, 5},
  {3, 3, 1}, {3, 3, 1}, {3, 3, 3}, {3, 3, 3}, {3, 3, 5}, {3, 5, 1}, {3, 5, 1}, {3, 5, 3}, {3, 5, 3}, {3, 5, 5},
  {3, 1, 1}, {3, 1, 1}, {3, 1, 3}, {3, 1, 3}, {3, 1, 5}, {3, 1, 1}, {3, 1, 1}, {3, 1, 3}, {3, 1, 3}, {3, 1, 5},
  {3, 3, 1}, {3, 3, 1}, {3, 3, 3}, {3, 3, 3}, {3, 3, 5}, {3, 3, 1}, {3, 3, 1}, {3, 3, 3}, {3, 3, 3}, {3, 3, 5},
  {3, 5, 1}, {3, 5, 1}, {3, 5, 3}, {3, 5, 3}, {3, 5, 5}, {5, 1, 1}, {5, 1, 1}, {5, 1, 3}, {5, 1, 3}, {5, 1, 5},
  {5, 1, 1}, {5, 1, 1}, {5, 1, 3}, {5, 1, 3}, {5, 1, 5}, {5, 3, 1}, {5, 3, 1}, {5, 3, 3}, {5, 3, 3}, {5, 3, 5},
  {5, 3, 1}, {5, 3, 1}, {5, 3, 3}, {5, 3, 3}, {5, 3, 5}, {5, 5, 1}, {5, 5, 1}, {5, 5, 3}, {5, 5, 3}, {5, 5, 5}}
```

- en este caso, no tiene sentido considerar una terna más de una vez; por ejemplo, {1,1,1}
  - **Strings[*list*,*r*]**. Construye todas las posibles variaciones con repetición de longitud r a partir de los n elementos de la lista dada como primer argumento. Es necesaria la carga del paquete `Combinatorica`.

Ejemplo. Obtención de todos los números de tres cifras que pueden escribirse con los dígitos del número 11.335. ¿Cuántas son?

- definición de la lista

```
In[9]:= numvr = {1, 1, 3, 3, 5};
```

- la carga del paquete `Combinatorica` se realizó anteriormente

```
(* Needs["Combinatorica`"] *)
```

- posibles variaciones con repetición

```
In[12]:= varrnum = Strings[numvr, 3]

Out[12]= {{1, 1, 1}, {1, 1, 3}, {1, 1, 5}, {1, 3, 1}, {1, 3, 3}, {1, 3, 5}, {1, 5, 1}, {1, 5, 3}, {1, 5, 5},
 {3, 1, 1}, {3, 1, 3}, {3, 1, 5}, {3, 3, 1}, {3, 3, 3}, {3, 3, 5}, {3, 5, 1}, {3, 5, 3}, {3, 5, 5},
 {5, 1, 1}, {5, 1, 3}, {5, 1, 5}, {5, 3, 1}, {5, 3, 3}, {5, 3, 5}, {5, 5, 1}, {5, 5, 3}, {5, 5, 5}}
```

- número de variaciones
  - contando el número de elementos de la lista **varrnum**

```
In[13]:= Length[varrnum]

Out[13]= 27
```

Combinaciones

Sin repetición

Combinaciones de n elementos tomados de r en r son los diferentes grupos que pueden formarse con los n elementos dados, tomados de r en r, de modo que dos grupos son distintos entre sí cuando, al menos, difieren en un elemento. No se tiene en cuenta el orden en que estén dispuestos los elementos.

- **Subsets[*list*,{*r*}]**. Genera una lista que contiene todos los subconjuntos formados, exactamente, por r elementos de la lista dada como primer argumento.

Cálculo: $\boxed{C_{n,r} = \binom{n}{r} = \dfrac{n\,!}{r\,!\cdot(n-r)\,!}}$

- **Binomial[*n*,*r*]**. Calcula el número combinatorio $\binom{n}{r}$.

Ejemplo. Obtención de todas las posibles combinaciones que pueden realizarse con cuatro cuadrados de diferentes colores tomados de dos en dos. ¿Cuántas son?

- definición de la lista

```
colorc = {Green, Red, Blue, Black}

{■, ■, ■, ■}
```

- posibles combinaciones

```
comcolor = Subsets[colorc, {2}]

{{■, ■}, {■, ■}, {■, ■}, {■, ■}, {■, ■}, {■, ■}}
```

- número de combinaciones
  - contando el número de elementos de la lista **comcolor**

```
Length[comcolor]
```

6

  - aplicando la fórmula

```
nc = Length[colorc]; rc = 2;
```

```
Binomial[nc, rc]
```

6

Ejemplo. Obtención de todas las combinaciones de tres cifras que pueden escribirse con los dígitos del número 12.345. ¿Cuántas son?

- definición de la lista

```
numc = {1, 2, 3, 4, 5}
```

{1, 2, 3, 4, 5}

- posibles combinaciones

```
comnum = Subsets[numv, {3}]
```

{{1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 4}, {1, 3, 5}, {1, 4, 5}, {2, 3, 4}, {2, 3, 5}, {2, 4, 5}, {3, 4, 5}}

- número de combinaciones
  - contando el número de elementos de la lista **comnum**

```
Length[comnum]
```

10

  - aplicando la fórmula

```
nc = Length[numc]; rc = 3;
```

```
Binomial[nc, rc]
```

10

  - **KSubsets[*l*,*k*]**. Devuelve una lista ordenada que contiene todos los subconjuntos formados, exactamente, por k elementos de la lista l dada como primer argumento. Es necesaria la carga del paquete Combinatorica.

Ejemplo. Hallar todas las posibles combinaciones formadas por dos letras que pueden realizarse con las letras a, b, c y d. ¿Cuántas son?

- definición de la lista

```
listaabcd = {a, b, c, d};
```

- la carga del paquete Combinatorica se realizó anteriormente

```
(* Needs["Combinatorica`"] *)
```

- posibles combinaciones

```
comabcd = KSubsets[listaabcd, 2]
```

{{a, b}, {a, c}, {a, d}, {b, c}, {b, d}, {c, d}}

- número de combinaciones
  - contando el número de elementos de la lista **comabcd**

```
Length[comabcd]
```

6

  - aplicando la fórmula

```
Binomial[Length[listaabcd], 2]
```

6

Con repetición

Combinaciones con repetición de n elementos tomados de r en r son los diferentes grupos que pueden formarse con los n elementos dados, tomados de r en r, en los que pueden aparecer elementos repetidos, de modo que dos grupos son distintos entre sí cuando, al menos, difieren en un elemento. No se tiene en cuenta el orden en que estén dispuestos los elementos.

Cálculo: $\boxed{CR_{n,r} = \binom{n+r-1}{r} = \frac{(n+r-1)!}{r!\cdot(n-1)!}}$

Una forma de obtener las combinaciones con repetición consiste en concatenar las funciones **Select**, **Tuples** y **Sort**.

  - **Select[*list,crit*]**. Selecciona todos los elementos de la lista dada como primer argumento que verifican el criterio indicado como segundo argumento.

Ejemplo. Hallar todas las posibles combinaciones con repetición formadas por dos letras que pueden realizarse con las letras a, b y c. ¿Cuántas son?

- definición de la lista

```
listaabc = {a, b, c};
```

- selección de las combinaciones posibles: entre todas las duplas que contienen las dos mismas letras se seleccionan las que están en orden alfabético

```
comrabc = Select[Tuples[Sort[listaabc], 2], OrderedQ]
```

{{a, a}, {a, b}, {a, c}, {b, b}, {b, c}, {c, c}}

- número de combinaciones con repetición
  - contando el número de elementos de la lista **comrabc**

```
Length[comrabc]
```

6

  - aplicando la fórmula

```
ncr = Length[listaabc]; rcr = 2;
Binomial[ncr + rcr - 1, rcr]
```

6

Ejemplo. Obtención de todas las combinaciones con repetición de tres cifras que pueden escribirse con los dígitos del número 12.345. ¿Cuántas son?

- definición de la lista

```
numc = {1, 2, 3, 4, 5}
```

{1, 2, 3, 4, 5}

- selección de las combinaciones posibles: entre todas las ternas que contienen las tres mismas cifras se seleccionan las que están ordenadas de menor a mayor

```
comrc = Select[Tuples[Sort[numc], n = 3], OrderedQ]
```

```
{{1, 1, 1}, {1, 1, 2}, {1, 1, 3}, {1, 1, 4}, {1, 1, 5}, {1, 2, 2}, {1, 2, 3}, {1, 2, 4},
 {1, 2, 5}, {1, 3, 3}, {1, 3, 4}, {1, 3, 5}, {1, 4, 4}, {1, 4, 5}, {1, 5, 5}, {2, 2, 2}, {2, 2, 3},
 {2, 2, 4}, {2, 2, 5}, {2, 3, 3}, {2, 3, 4}, {2, 3, 5}, {2, 4, 4}, {2, 4, 5}, {2, 5, 5}, {3, 3, 3},
 {3, 3, 4}, {3, 3, 5}, {3, 4, 4}, {3, 4, 5}, {3, 5, 5}, {4, 4, 4}, {4, 4, 5}, {4, 5, 5}, {5, 5, 5}}
```

- número de combinaciones con repetición
  - contando el número de elementos de la lista **comrc**

```
Length[comrc]
```

```
35
```

  - aplicando la fórmula

```
ncr = Length[numc]; rcr = 3;
Binomial[ncr + rcr - 1, rcr]
```

```
35
```

Otra forma de obtener todas las combinaciones con repetición consiste en concatenar las funciones **Tuples**, **Sort**, **Map** y **DeleteDuplicates**.

  - **Map[*f*,*expr*]**. Aplica la función *f* a cada elemento del primer nivel de *expr*.
  - **DeleteDuplicates[*list*]**. Elimina todos los elementos duplicados que se encuentran en la lista *list*.

Ejemplo. Hallar todas las posibles combinaciones con repetición formadas por dos letras que pueden realizarse con las letras *a*, *b* y *c*.

- definición de la lista

```
listaabc = {a, b, c};
```

- selección de las combinaciones posibles: todas las duplas se ordenan alfabéticamente y se eliminan las duplicadas

```
comrabc = DeleteDuplicates[Map[Sort, Tuples[listaabc, 2]]]
```

```
{{a, a}, {a, b}, {a, c}, {b, b}, {b, c}, {c, c}}
```

Ejemplo. Obtención de todas las posibles combinaciones con repetición que pueden realizarse con tres cuadrados de diferentes colores.

```
DeleteDuplicates[Map[Sort, Tuples[{Red, Blue, Green}, 3]]]
```